Loan Amount Prediction Documentation

1.Descriptive Analysis of All Variable

Data Overview

The initial steps in the code involve loading the dataset and performing basic exploratory data analysis (EDA) to understand the structure and distribution of the variables.

1] **df.head():** Displays the first few rows of the dataset to get a sense of what the data looks like.
2] **df.describe()**: Provides a summary of the statistics for numerical variables.
3] **df.dtypes** and **df.info():** Offer insights into the data types and the structure of the dataset.
4] **df.isnull()** : Checks for missing values in the dataset.

Descriptive Analysis of Financial, House, and Loan Details
The dataset is divided into three separate DataFrames to focus on specific areas:
1] **financial_details_df**`: Contains financial-related columns such as business type, income, and expenses.
2] **house_details_df** : Contains house-related columns like ownership, type, and area.
3] **loan_details_df** : Contains loan-related information such as purpose, tenure, and amount.

Numerical and Categorical Summary

The code separates numerical and categorical variables for a more focused analysis:
1] **df_numeric** = df.select_dtypes(include=[np.number]) : Extracts only the numerical columns for further analysis.
2] **numerical_summary** and **categorical_summary** : Provide statistical summaries for numerical and categorical data, respectively.

Visual Analysis:

Histograms and correlation matrices are used to visualise the distributions and relationships between variables:
1] **df.hist(figsize=(12, 12))** : Displays histograms for all numerical variables.
2] **sns.heatmap**(numerical_df.corr(), annot=True, cmap='coolwarm', fmt='.2f'): Shows the correlation matrix of the numerical variables.
3] **sns.countplot**: Plots the distribution of categorical variables.

Missing Values

Missing Data Analysis:

`df.isnull().sum()` counts missing values in each column. This is crucial for deciding on imputation strategies.

Handling Missing Data

Imputation Strategy:

Simple Imputer: Used to fill in missing values with the mean of the column. This ensures that no data points are discarded and that the imputation maintains the overall distribution.

Implementation:

```
from sklearn.impute import SimpleImputer
imputer = SimpleImputer(strategy='mean')
X_train = imputer.fit_transform(X_train)
X_test = imputer.transform(X_test)
```

Role: Imputers are used to ensure that missing values do not lead to errors during model training. The mean imputation strategy replaces missing values with the average value of the column, maintaining the integrity of the dataset and allowing the model to learn from all available data.

2. Best Models for Predicting Loan Amount

Model Selection

The code implements several regression models to predict the `loan_amount`:

1. Random Forest Regressor
2. Gradient Boosting Regressor
3. Linear Regression
4. Logistic Regression

Model Performance

The models are evaluated using the following metrics:
Root Mean Squared Error (RMSE)
R-squared (R^2)
Mean Absolute Error (MAE)
Mean Squared Error (MSE)

Best Model Selection

Based on the performance metrics:

Random Forest Regressor and Gradient Boosting Regressor are the most suitable models for predicting loan amounts, given their high R^2 scores and low RMSE values.

3.Building a Model to Predict Maximum Loan Amount

Feature Selection

The code leverages `LabelEncoder` to encode categorical variables and uses all features (`X`) except for `Id` and `loan_amount` as predictors.

`X` (Feature Matrix): This is a matrix containing all the features (independent variables) that your model will use to make predictions. Each row corresponds to a different data point, and each column corresponds to a different feature. In your case, `X` is the dataset with all the relevant features except for the target variable (`loan_amount`) and possibly an identifier column (`Id`), which doesn't carry predictive power.

`y` (Target Variable): This is the variable you want to predict (dependent variable). In your case, `y` is the `loan_amount` column, which represents the loan amount you want to predict based on the features in `X`.

Scaling

The features are scaled using `StandardScaler`, which standardise the data to have a mean of 0 and a standard deviation of 1. This is important for many machine learning algorithms, especially those involving distance-based measures.

Model Training and Prediction

Models are trained using `train_test_split` to split the data into training and testing sets. The models are then evaluated based on their predictions.

4. Significance of `loan_purpose` as a Predictor

Loan purpose is significant with p-value: 8.494523336913686e-08

Include `loan_purpose` as per business requirements,ensuring it doesn’t degrade overall model performance.

5. Measuring the Fitness of the Model

RMSE: Measures the average magnitude of the error.

R^2: Indicates how well the model explains the variance in the target variable.

MAE: Provides the average absolute difference between predicted and actual values.

MSE: Measures the average of the squares of the errors.

Classification Models

Accuracy: The proportion of correct predictions.

Confusion Matrix: Provides insight into the types of errors made.

Classification Report: Includes precision, recall, and F1-score, offering a more detailed view of model performance.